Normalized Texture Motifs and Their Application to Statistical Object Modeling

S. D. Newsam

March 10, 2004

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Normalized Texture Motifs and Their Application to Statistical Object Modeling

Abstract

*A fundamental challenge in applying texture features to statistical object modeling is recognizing differently oriented spatial patterns. Rows of moored boats in remote sensed images of harbors should be consistently labeled regardless of the orientation of the harbors, or of the boats within the harbors. This is not straightforward to do, however, when using anisotropic texture features to characterize the spatial patterns. We here propose an elegant solution, termed **normalized texture motifs**, that uses a parametric statistical model to characterize the patterns regardless of their orientation. The models are learned in an unsupervised fashion from arbitrarily orientated training samples. The proposed approach is general enough to be used with a large category of orientation-selective texture features.*

1. Introduction

The human visual system is adept at recognizing common spatial patterns. It is easy to recognize that the rows of moored boats and water are the common patterns in aerial images of harbors, and then use this knowledge to discover other harbor-like regions. Automating this, however, is not easy. Texture features can be used to characterize the spatial patterns. But most objects for which a texture-based modeling approach is appropriate 1) do not have identifiable orientations, and 2) contain patterns that do not always occur at the same orientation. As a result, the texture features corresponding to the common patterns will not be clustered in the feature space. This represents a significant challenge.

This work introduces *normalized texture motifs*, a novel solution to the problem of using texture features for statistical object modeling. The proposed framework has two major benefits. First, only one orientation of each spatial pattern is characterized. This greatly simplifies the model. The second benefit is that no constraints are placed on the training images. An unsupervised learning algorithm is used to estimate the model parameters from training sets that need not contain all orientations of a spatial pattern. These advantages follow from a key insight into the structure of texture features extracted using scale- and orientation-selective filtering.

2. Related Work

There are generally two approaches to analyzing image texture in a rotationally invariant way. The first is to either extract rotationally invariant features [1][2][3], or make them invariant through post-processing [4][5][6]. Extracting isotropic features dismisses the issue of rotation invariance from the outset. The discriminating power of these features is limited, however, since they disregard orientation, a fundamental property of texture. Approaches that make the features rotation invariant through post-processing are also problematic. In some situations, the relative orientation between texture components at different scales is lost. In other situations, the features are made rotationally invariant with respect to a dominant direction only.

The second approach to analyzing texture in a rotationally invariant way is to incorporate the rotational invariance into the learning or classification stage. The proposed technique falls into this category. These approaches typically use spatial proximity as one of the criteria for grouping related textures, and, therefore, are not appropriate for modeling objects in which the same pattern occurs at different locations.

The proposed approach is novel in that it 1) considers texture orientation, 2) preserves correspondence between the orientation at different scales, 3) incorporates rotation invariance into the clustering step, and 4) is appropriate for modeling objects composed of differently located and orientated patches.

The proposed approach achieves orientation invariance using methods similar to those used in [7] to group image and video frames that have undergone global transformations, such as translation. The proposed approach instead uses the techniques to compensate for the effects that *local transformations* have on a *feature space*. The proposed approach also addresses a different problem, that of using texture to model complex objects.

3. Image Texture Features

The proposed approach assumes that texture feature vectors extracted using spatial filters tuned to combinations of R orientations and S scales are used to characterize the spatial

patterns:

$$x^n = [(x_{11}^n, x_{12}^n, \ldots, x_{1S}^n), (x_{21}^n, x_{22}^n, \ldots, x_{2S}^n), \\ \ldots, (x_{R1}^n, x_{R2}^n, \ldots, x_{RS}^n)] . \quad (1)$$

Here, element x_{rs}^n represents the output at pixel location n of the filter tuned to orientation r and scale s. Parentheses are used to group the filter outputs by orientation for clarity. A typical filter bank consists of filters tuned at $180/R$ degree intervals.

Fundamental to the proposed approach is the fact that even though these texture features are orientation selective, they can be used to enable rotation invariant applications. Rotation invariant similarity retrieval is possible, for example, by using a modified distance function,

$$d_{RI}(x^1, x^2) = \min_{r \in R} \left\| x_{<r>}^1 - x^2 \right\|_2 \quad (2)$$

where $x_{<r>}^n$ represents x_n circularly shifted by r orientations:

$$x_{<r>}^n = \quad (3) \\ [(x_{r1}^n, x_{r2}^n, \cdots, x_{rS}^n), (x_{(r+1)1}^n, x_{(r+1)2}^n, \cdots, x_{(r+1)S}^n), \\ \cdots, (x_{R1}^n, x_{R2}^n, \cdots, x_{RS}^n), (x_{11}^n, x_{12}^n, \cdots, x_{1S}^n), \\ \cdots, (x_{(r-1)1}^n, x_{(r-1)2}^n, \cdots, x_{(r-1)S}^n)] .$$

Conceptually, this distance function computes the best match between rotated versions of the images.

4. Texture Motifs

A texture motif is here defined as *a characteristic spatial pattern common to a class of objects*. The pattern can occur at different locations and orientations within the objects. Examples of texture motifs include the rows of moored boats in aerial images of harbors (as shown in Figure 2(a)) and the rows of trees in aerial images of golf courses. Accurate characterizations of texture motifs can facilitate automated object recognition. However, developing the characterizations is a challenge largely due to the high-dimensionality of the texture feature spaces. In the proposed approach, the distribution of the feature vectors corresponding to a texture motif is characterized with a parametric statistical model whose parameters are estimated from unlabeled training samples using an unsupervised learning algorithm. The advantages of this approach are 1) preprocessing steps, such as segmentation are not required; 2) the learning phase is completely automated, only requiring unlabeled training examples; and 3) the model form is fixed so that only the parameter values differ from the model of one class of objects to another.

4.1. Statistical Modeling via GMMs

The feature vectors corresponding to a texture motif are assumed to have a Gaussian distribution in the RS dimensional texture feature space. The conditional probability a feature vector x, given that it is generated by texture motif j, is thus

$$p(x|j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j (x-\mu_j)} \quad (4)$$

where $d = RS$. The density for motif j is completely specified by the parameters (μ_j, Σ_j), where μ_j is the mean vector and Σ_j is the covariance matrix. The density of an ensemble of J texture motifs is consequently modeled using a Gaussian Mixture Model (GMM), so that the unconditional probability of a feature vector x, with respect to an ensemble of motifs, can be computed as

$$p(x) = \sum_{j=1}^{J} P(j) p(x|j) \quad (5)$$

where $P(j)$ is the prior probability of motif j and $p(x|j)$ is the conditional probability. The model for an ensemble of J motifs is completely specified by the parameters

$$\Theta = \{\theta_j = (P(j), \mu_j, \Sigma_j) | j = 1 \ldots J\} . \quad (6)$$

4.2. Unsupervised Learning via the EM Algorithm

The Expectation Maximization (EM) algorithm [8] is a common technique for learning the parameters of a statistical model from an unlabeled training set. It is an iterative learning technique in which the observable data is augmented by the missing elements needed to estimate the values of the model parameters. The missing elements for GMMs are the mixture assignments of the feature vectors, $z \in J$. Estimating the parameter values $\{\theta_j | j = 1 \ldots J\}$ from the training set is straightforward if these assignments are known (using maximum-likelihood, for example). The complete data for the GMM case is $y = (x, z)$, where x is the observable feature vector and z is the unknown mixture assignment.

Each iteration m of the EM procedure has two steps. First, the current estimates of the parameters are used to compute the expected value of the log-likelihood of the complete training data given the observed data where the expectation is with respect to the unknown data. Second, the parameter estimates are updated to maximize this expectation. These two steps are repeated until a stopping criterion is met, often related to the rate of change of the likelihood.

For the GMM case, the current estimates at iteration m are $\Theta^{(m)} = \{\theta_j^{(m)} | j = 1 \ldots J\}$; the complete data are

$Y = (X, Z) = \{(x^n, z^n) | n = 1 \ldots N\}$; the observed data are $X = \{x^n | n = 1 \ldots N\}$; and the unknown data are $Z = \{z^n | n = 1 \ldots N\}$, where N is the size of the training set. The expected value of the log-likelihood of the training data is commonly written as

$$Q\left(\Theta | \Theta^{(m)}\right) = E_Z\left[\log p\left(Y | \Theta\right) | X, \Theta^{(m)}\right] \, . \quad (7)$$

The maximization step involves updating the parameter estimates to the values of Θ that maximize Q. Fortunately, this optimization problem has an analytical solution for the GMM case, and the updated estimates for the model parameters are computed as

$$P^{(m+1)}\left(j\right) = \frac{1}{N} \sum_{n=1}^{N} P^{(m)}\left(j | x^n\right) , \quad (8)$$

$$\mu_j^{(m+1)} = \frac{\sum_{n=1}^{N} P^{(m)}\left(j | x^n\right) x^n}{\sum_{n=1}^{N} P^{(m)}\left(j | x^n\right)} , \quad (9)$$

and

$$\Sigma_j^{(m+1)} = \quad (10)$$

$$\frac{\sum_{n=1}^{N} P^{(m)}\left(j | x^n\right) \left(x^n - \mu_j^{(m+1)}\right) \left(x^n - \mu_j^{(m+1)}\right)^T}{\sum_{n=1}^{N} P^{(m)}\left(j | x^n\right)} \, .$$

The k-means clustering algorithm is typically used to initialize the parameter values, $\Theta^{(0)}$.

5. Normalized Texture Motifs

Modeling a motif with a single mixture component is problematic since the feature vectors depend on the orientation of the texture. The feature vectors corresponding to one orientation of moored boats might be clustered in the high-dimensional feature space, but the feature vectors corresponding to multiple orientations cannot assumed to be. A potential solution is to model a motif with multiple components, one for each orientation at which the motif might occur. However, this not only increases the complexity of the model and the number of parameters that must be learned but also requires that the training set contain a sufficient number of samples of all possible orientations of the motif.

The major contribution of this paper is a method that uses GMMs and the EM algorithm to model texture motifs *without increasing the complexity of the model or the number of parameters that must be learned*. Furthermore, *no constraints are placed on the orientations of the motif samples in the training set*. This generalization is achieved by exploiting the structure of the texture feature vectors in a manner inspired by the rotation invariant distance function.

5.1. The Normalized Statistical Model

The feature vectors corresponding to a single orientation of a texture motif, termed the normalized orientation, are now assumed to be Gaussian distributed. The conditional probability of a feature vector x, given that it is generated from texture motif j at orientation r, is thus

$$p\left(x | j, r\right) = \frac{1}{\left(2\pi\right)^{d/2} \left|\Sigma_j\right|^{1/2}} \, e^{-\frac{1}{2}\left(x_{\langle -r \rangle} - \mu_j\right)^T \Sigma_j \left(x_{\langle -r \rangle} - \mu_j\right)} \, .$$
$$(11)$$

The orientation r means with respect to the normalized orientation. The term $x_{<-r>}$ corresponds to the vector x circularly shifted by $-r$ orientations (Eq. 4). Conceptually, this corresponds to un-rotating the feature vector to the normalized orientation of the motif.

The density for motif j is still completely specified by the parameters (μ_j, Σ_j). The density of an ensemble of J texture motifs is again modeled using a GMM, but now the unconditional probability of a feature vector x, with respect to an ensemble of motifs, is

$$p\left(x\right) = \sum_{j=1}^{M} \sum_{r=1}^{R} p\left(x | j, r\right) P(j) P(r) \, . \quad (12)$$

The orientation of a feature vector is assumed to be independent of its motif assignment. All rotations are considered equiprobable:

$$P(r) = \frac{1}{R} \text{ for } r = 1 \ldots R \, . \quad (13)$$

The complexity of the model has not increased since an ensemble of J motifs is still completely specified by the parameters $\Theta = \{\theta_j = (P(j), \mu_j, \Sigma_j) | j = 1 \ldots J\}$.

5.2. Normalized Unsupervised Learning

The EM algorithm is again used to estimate the parameters of the normalized motif model from the unlabeled training sets, but now the missing elements are not only the motif assignment $z \in J$ of an observation but also its *orientation* $v \in R$ with respect to the normalized orientation of the motif. Again, estimating the parameter values would straightforward if the motif assignments and orientations were known. The complete data for each observation is now $y = (x, z, v)$.

For the normalized learning case, the current estimates at iteration m are still $\Theta^{(m)} = \{\theta_j^{(m)} | j = 1 \ldots J\}$; but the complete data are now $Y = (X, Z, V) = \{(x^n, z^n, v^n) | n = 1 \ldots N\}$ where the unknown data are $(Z, V) = \{(z^n, v^v) | n = 1 \ldots N\}$. The expected value of the log-likelihood of the training data is now computed with respect to both Z and V:

$$Q\left(\Theta | \Theta^{(m)}\right) = E_{Z,V}\left[\log p\left(Y | \Theta\right) | X, \Theta^{(m)}\right] \, . \quad (14)$$

Q can be expanded as:

$$
\begin{aligned}
Q &= E_{Z,V}\left[\sum_{n=1}^{N}\log p\left(x^n, z^n, v^n|\Theta\right)|X,\Theta^{(m)}\right] \\
&= \sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{n=1}^{N}\log p\left(x^n, z^n, v^n|\Theta\right) \qquad (15) \\
&\quad \cdot P\left(z^n = j, v^n = r|X,\Theta^{(m)}\right).
\end{aligned}
$$

The joint probability of the complete data can be expanded using Bayes' rule and the independence of z and v:

$$
\begin{aligned}
p\left(x^n, z^n, v^n|\Theta\right) &= p\left(x^n|z^n, v^n, \Theta\right)P\left(z^n, v^n|\Theta\right) \quad (16) \\
&= p\left(x^n|z^n, v^n, \Theta\right)P\left(z^n|\Theta\right)P\left(v^n\right).
\end{aligned}
$$

Thus, Q becomes

$$
\begin{aligned}
Q &= \sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|X,\Theta^{(m)}\right) \\
&\quad \cdot \{\log p\left(x^n|z^n = j, v^n = r, \Theta\right) \qquad (17) \\
&\quad + \log P\left(z^n = j|\Theta\right) + \log P\left(v^n = r\right)\}.
\end{aligned}
$$

The maximization step computes the estimate of the parameters for the next iteration of the algorithm as

$$
\Theta^{(m+1)} = \arg\max_{\Theta} Q\left(\Theta|\Theta^{(m)}\right). \qquad (18)
$$

Fortunately, this optimization problem also has an analytical solution, and the updated estimates for the model parameters are computed as

$$
P^{(m+1)}(j) = \frac{1}{N}\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|x^n, \Theta^{(m)}\right), \qquad (19)
$$

$$
\mu_j^{(m+1)} = \frac{\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|x^n, \Theta^{(m)}\right)x_{<-r>}^n}{\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|x^n, \Theta^{(m)}\right)}, \qquad (20)
$$

and

$$
\Sigma_j^{(m+1)} = \frac{\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|x^n, \Theta^{(m)}\right)A_{r,n,j}^{(m+1)}}{\sum_{r=1}^{R}\sum_{n=1}^{N}P\left(z^n = j, v^n = r|x^n, \Theta^{(m)}\right)} \qquad (21)
$$

where

$$
A_{r,n,j}^{(m+1)} = \left(x_{<-r>}^n - \mu_j^{(m+1)}\right)\left(x_{<-r>}^n - \mu_j^{(m+1)}\right)^T. \qquad (22)
$$

The initial estimate of the parameter values $\Theta^{(0)}$ is computed using a modified k-means clustering algorithm in which 1) the Euclidean distance is replaced with the rotation invariant distance measure, and 2) the best circularly shifted versions of cluster members are used to update the cluster centroids.

6. Experimental Results

The proposed technique for modeling objects composed of multiple spatial patterns is compared to two alternate methods, using both synthetic and real datasets.

6.1. The Four Methods

Method 1 is the proposed modeling and learning technique, normalized texture motifs.

Method 2 uses GMMs and the EM algorithm to model the motifs *without accounting for the texture orientation*. This corresponds the method described in Section 4. This method is expected to have difficulty using a single mixture component to represent all orientations at which a motif might occur.

Method 3 includes a pre-processing step in which each feature vector is circularly shifted in an independent fashion so that its maximum average energy is at orientation 0 (the average is taken over the scales). This is a common approach that results in the feature vectors being rotationally invariant, at least with respect to the dominant orientation, if one exists. GMMs and the EM algorithm are then used to model the distribution of the shifted features. While this pre-processing should result in more *compact* feature clusters, this grouping might not be as perceptually satisfying as the one produced by the proposed approach since it considers all the features.

Texture feature vectors are extracted using a bank of Gabor filters tuned to $R=6$ orientations and $S=5$ scales. Note, however, that the proposed approach is general enough to use any descriptor characterized by scale- and orientation-selective filtering.

6.2. The Datasets

The three methods are used to model the texture motifs in synthetic images created using the Brodatz texture dataset [9]. Five images are created for each pair of textures by placing randomly sized and oriented patches of one of the textures on a background composed of the another texture, also at random orientations.

The three methods are also used to model the texture motifs in a set of five aerial images of harbor regions. Harbors are one example of the kinds of objects appearing in remote sensed imagery that consist of spatial patterns at various, disconnected orientations. Some other examples include golf courses, mobile home parks, and vineyards.

4

6.3. The Models and Measures

One way to evaluate a model is to observe how well it classifies a novel instance of an object. We do this here using a leave-one-out approach in which the model is trained on all but one of the images in a set, and then used to classify the remaining, test image. In particular, the learned model is used to assign a motif label to each pixel in the test image using a maximum a priori (MAP) classifier:

$$\text{motif} = \arg\max_{1 \leq j \leq J} P\left(j|x\right) \ . \tag{23}$$

The models learned using the proposed approach can also be used to estimate the orientation of the texture at each pixel, again using a MAP classifier:

$$\text{orientation} = \arg\max_{1 \leq r \leq R} P\left(r|x\right) \ . \tag{24}$$

The images in the simulated dataset are modeled using GMMs with two mixture components. The number of components in the GMMs used to model the real images is manually chosen. We have had some success in using a minimum description criterion to automatically chose the model order but this challenging problem is beyond the scope of this paper.

6.4. Classification Results

Figure 1(b) shows the MAP motif assignments for the synthetic image in Figure 1(a) using the proposed approach. Note that the two textures are consistently labeled, regardless of their orientations. Figure 1(c) shows the assignments using method 2, which does not account for orientation. Note that the two textures are incorrectly labeled as the same motif. These results are similar to those for other texture pairs.

Figure 1(d) shows the MAP orientation assignments using the proposed approach. Note that the assignments are consistent both within and between the uniformly textured regions. Figure 1(e) shows the orientation assignments that result from the pre-processing in method 3. Note that the assignments are not always consistent within the uniformly textured regions. This is because the orientation of the texture at each pixel is estimated independently, without knowledge of the other pixels in the image.

Figure 2(b) shows the motif assignments for the harbor image in Figure 2(a) using method 2, which does not account for orientation. Note that moored boats at different orientations are incorrectly labeled as different motifs. Compare this with Figure 3(a) which shows the motif assignments using the proposed approach. Different motif labels are now assigned to the different densities of the boats rather than the different orientations. This labeling is more reasonable from a perceptual standpoint. Finally, Figure 3(b) shows the orientation assignments for the proposed approach. Knowledge of the orientation of the motifs is shown to be preserved. Such information might be useful if, for example, the relative orientations of the motifs were found to be an important characteristic of an object. The results shown here for harbor regions are similar to those produced for other objects, such as mobile home parks and golf courses.

7. Conclusion

This work presents normalized texture motifs, a novel approach to learning the spatial patterns common to a set of objects. Rotation invariance is incorporated into an unsupervised learning algorithm by exploiting the properties of texture descriptors characterized by scale- and orientation-selective filtering. The method is appropriate for use with a wide range of texture features, such as those extracted using Gabor filters. Experimental results show normalized texture motifs outperform alternate approaches for both synthetic and real images.

Future work includes using a similar approach to account for variations in scale. This would allow the models to be independent, within limits, of the resolution of the images. A single model could be used, for example, to recognize harbors in remote sensed images with different, and, possibly unknown, spatial resolutions.

References

[1] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, volume 2, pages 39–45, 2001.

[2] L. Wang and G. Healey. Using Zernike moments for the illumination and geometry invariant classification of multispectral texture. *IEEE Trans. on IP*, 7(2):196–203, 1998.

[3] M. Pietikainen, T. Ojala, and Z. Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[4] H. Greenspan, S. Belongie, R. Goodman, and P. Perona. Rotation invariant texture recognition using a steerable pyramid. In *ICPR*, volume 2, pages 162–167, 1994.

[5] G. Haley and B. S. Manjunath. Rotation-invariant texture classification using modified Gabor filters. In *ICIP*, volume 1, pages 262–265, 1995.

[6] M. M. Leung and A. M. Peterson. Scale and rotation invariant texture classification. In *IEEE Asilomar Conf. on Signals, Systems and Computers*, volume 1, pages 461–465, 1992.

[7] B. J. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(1):1–17, 2003.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *J. of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, U.S.A., 1966.
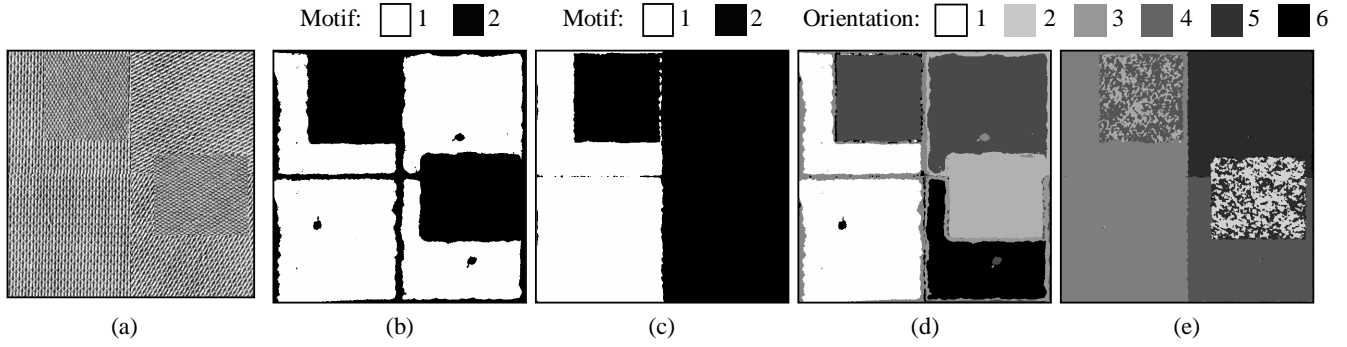
Figure 1: (a) An example image from the synthetic dataset. (b) Motif assignments using the proposed approach. The two textures are consistently labeled, regardless of their orientation. (c) Motif assignments using an alternate approach (method 2). The different textures are incorrectly labeled as the same motif. (d) Orientation assignments using the proposed approach. The assignments are consistent both within and between the uniformly textured regions. (e) Orientation assignments using an alternate approach (method 3). The assignments are not consistent within the uniformly textured regions since the orientation of each pixel is estimated independently, without knowledge of the other pixels.
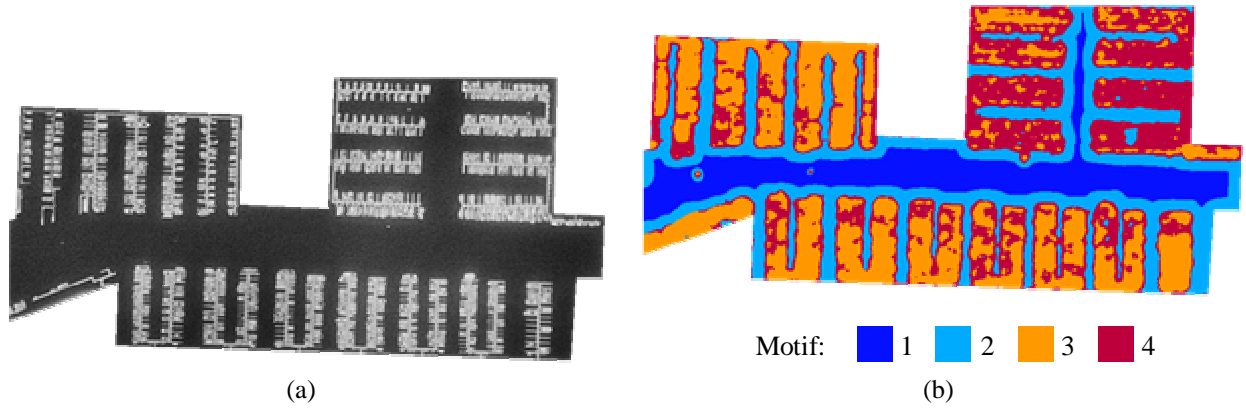

Figure 2: (a) An example of a harbor. (b) Motif assignment using an approach that does not account for orientation (method 2). The moored boats at different orientations are incorrectly labeled as different motifs.
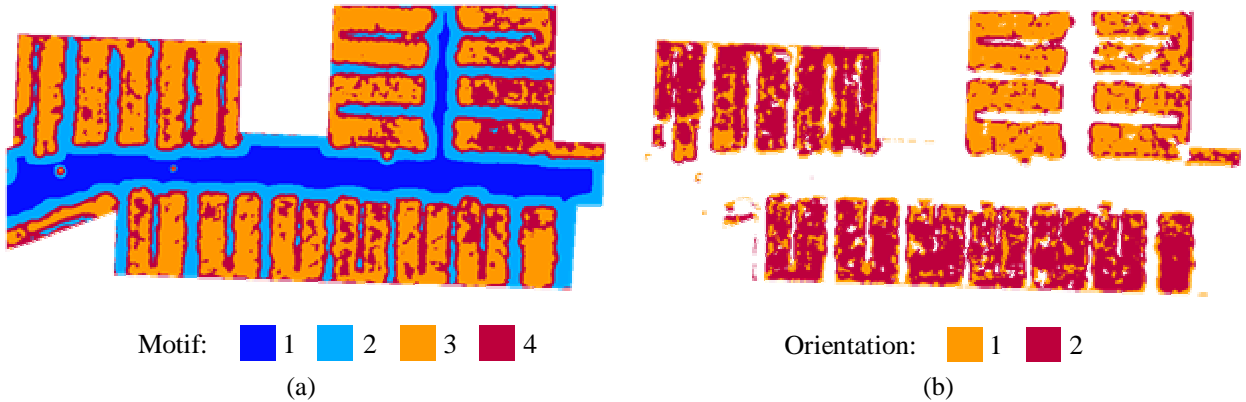

Figure 3: (a) Motif assignment using the proposed approach. The moored boats at different orientations are correctly labeled as the same motif. It is the different densities of boats that are labeled as different motifs. (b) Orientation assignment using the proposed approach. Knowledge of the orientation of the motifs is preserved.